

# Présentation sur UNLocBoX

Toolbox d'optimisation non-lisse

Travaux Pratiques Encadrés (TPE) – INF4127 : Optimisation 2

Nom	Prénom	Matricule
ESSUTHI MBANGUE	ANGE ARMEL	24F2456
TAGNE TALLA	IDRISS CHANEL	19M2351
DJATCHE-NKAMGANG	SYLVANO	22W2163
GOUJOU GUIMATSA	ZIDANE	21T2899

Université de Yaoundé I  
Département d'Informatique

*Sous l'encadrement du Professeur : Pr. MELATAGIA*

Année académique 2024–2025

# Plan de la présentation

- 1 Introduction
- 2 Contexte Mathématique
- 3 Structure de UNLocBoX
- 4 Solveurs disponibles
- 5 Définition des fonctions
- 6 Exemple pratique
- 7 Installation et utilisation
- 8 Avantages et limitations
- 9 Conclusion

# Introduction à UNLocBoX

## Qu'est-ce que UNLocBoX ?

- **UNLocBoX** (UNconstrained L0C convex Optimization toolBOX)
- Boîte à outils **MATLAB** pour l'optimisation convexe non lisse
- **Compatible GNU Octave** - alternative open source
- Implémente les méthodes de splitting proximal
- Développé à l'EPFL (Lausanne)
- Version Python disponible : **PyUNLocBoX**

## Problèmes adressés

Résolution de problèmes de la forme :

$$\min_{x \in \mathbb{R}^N} \sum_{n=1}^K f_n(x)$$

où les  $f_n$  sont des fonctions convexes, éventuellement non-différentiables.

## Pourquoi les méthodes proximales ?

- Adaptées aux problèmes à grande échelle (*big data*)
- Complexité par itération réduite (souvent  $O(N)$ )
- Gèrent les fonctions non-différentiables
- Convergence garantie sous conditions

## Exemple d'application

- Reconstruction d'images
- Apprentissage machine
- Traitement du signal
- Vision par ordinateur

# Opérateurs Proximaux

## Définition

Pour une fonction convexe  $f$ , l'opérateur proximal est défini par :

$$\text{prox}_{\lambda f}(x) = \arg \min_{y \in \mathbb{R}^N} \frac{1}{2} \|x - y\|_2^2 + \lambda f(y)$$

## Interprétation

- Généralisation de la projection
- Compromis entre minimiser  $f$  et rester proche de  $x$
- Unique solution (convexité stricte)

## Exemples connus

- Norme  $\ell_1$  : seuillage doux
- Norme  $\ell_2$  : projection sur la boule
- Contraintes : projection sur l'ensemble

# Structure de UNLocBoX

## Composition de la toolbox

- **Solvers** : Cur de la toolbox (FISTA, Douglas-Rachford, etc.)
- **Opérateurs proximaux** : Implémentation des prox pour diverses fonctions
- **Fonctions de démonstration** : Exemples d'utilisation
- **Utilitaires** : Fonctions auxiliaires

## Fonction principale : solvep

- Sélection automatique du solveur adapté
- Gestion des paramètres optionnels
- Retourne la solution et des informations de convergence
- **Code compatible MATLAB et GNU Octave**

## Exemple d'appel

```
sol = solvep(x0, {f1, f2, f3}, param);
```

# Solveurs disponibles

## Solveurs spécifiques (2 fonctions)

- **Forward-Backward** (FISTA) : 1 fonction différentiable + 1 prox
- **Douglas-Rachford** : 2 prox
- **ADMM** : 2 prox
- **Chambolle-Pock** : primal-dual

## Solveurs généraux (K fonctions)

- **Generalized Forward-Backward** : K prox
- **PPXA** : K prox
- **SDMM** : K prox

## Choix du solveur

La fonction `solvp` choisit automatiquement le solveur le plus adapté en fonction des fonctions fournies.

# Définition des fonctions

Chaque fonction  $f_k$  est représentée par une structure MATLAB avec des champs spécifiques :

## Fonction différentiable

- eval : évaluation de la fonction
- grad : gradient de la fonction
- beta : constante de Lipschitz du gradient

## Fonction non-différentiable

- eval : évaluation de la fonction
- prox : opérateur proximal

## Exemple de fonction - Compatible MATLAB/Octave

```
f1.eval = @(x) norm(A*x - y)^2;  
f1.grad = @(x) 2*A'*(A*x - y);  
f1.beta = 2*norm(A)^2;
```



# Opérateurs proximaux implémentés

## Normes courantes

- Norme  $\ell_1$  : `prox_l1`
- Norme  $\ell_2$  : `prox_l2`
- Norme  $\ell_{1,2}$  : `prox_l12`
- Norme nucléaire : `prox_nuclearnorm`
- Norme TV : `prox_tv`

## Projections

- Projection sur la boule  $\ell_2$  : `proj_b2`
- Projection sur la boule  $\ell_\infty$  : `proj_linf`
- Projection sur le simplexe : `proj_simplex`

## Autres fonctions

- Indicateur d'ensemble convexe
- Fonctions quadratiques

# Exemple pratique : reconstruction d'image

## Problème

Reconstruction d'une image  $x$  à partir de mesures bruitées  $y = Ax + n$  avec un prior de régularité TV.

## Formulation

$$\min_x \|Ax - y\|_2^2 + \lambda \|x\|_{TV}$$

## Décomposition

- $f_1(x) = \|Ax - y\|_2^2$  (différentiable)
- $f_2(x) = \lambda \|x\|_{TV}$  (non-différentiable)

# Code MATLAB/Octave pour l'exemple

## Définition des fonctions

```
% Paramètres
```

```
lambda = 0.1;
```

```
A = @(x) x; % Opérateur identité (exemple)
```

```
At = @(x) x;
```

```
% Fonction de fidélité aux données
```

```
f1.eval = @(x) norm(A(x) - y, 'fro')^2;
```

```
f1.grad = @(x) 2*At(A(x) - y);
```

```
f1.beta = 2*norm(A)^2;
```

```
% Régularisation TV
```

```
param_tv.verbose = 1;
```

```
param_tv.maxit = 50;
```

```
f2.eval = @(x) lambda*tv_norm(x);
```

```
f2.prox = @(x,T) prox_tv(x, lambda*T, param_tv);
```

# Code MATLAB/Octave (suite)

## Résolution

```
% Paramètres du solveur
param.verbose = 1;
param.maxit = 100;
param.tol = 1e-5;
x0 = zeros(size(y));
sol = solvep(x0, {f1, f2}, param);
```

## Résultat

La solution `sol` est l'image reconstruite avec une régularisation TV.

# Installation et utilisation

## Environnements supportés

- **MATLAB** (versions récentes) - Environnement natif
- **GNU Octave** - Alternative open source compatible

## Téléchargement

- <https://github.com/epfl-lts2/unlocbox>
- Ou <https://lts2.epfl.ch/unlocbox/>

## Dépendances

- MATLAB ou GNU Octave
- LTFAT (optionnel, pour certaines démos)
- GSPBox (optionnel, pour les graphes)

## Initialisation

Ajouter le chemin de UNLocBoX et exécuter `init_unlocbox.m`

# Avantages et limitations

## Avantages

- Interface simple et intuitive
- Large choix de solveurs et d'opérateurs
- Documentation complète
- Code open source
- Efficace pour les problèmes à grande échelle
- **Compatible avec MATLAB et GNU Octave**

## Limitations

- Nécessite une connaissance basique de l'optimisation
- Moins adapté aux problèmes très structurés (comme les SDP)
- Performance en MATLAB/Octave pure (peut être lent pour certains problèmes)

# Conclusion

## Résumé

- UNLocBoX est une toolbox **MATLAB/GNU Octave** pour l'optimisation convexe non lisse
- Elle utilise les méthodes de splitting proximal
- Elle offre une grande flexibilité dans la définition des problèmes

## Points clés

- Structure modulaire : solveurs, prox, démos
- Sélection automatique des solveurs
- Nombreux opérateurs proximaux implémentés
- **Code compatible avec MATLAB et GNU Octave**

## Perspectives

- Version Python (PyUNLocBoX)
- Intégration avec d'autres toolboxes (GSPBox, LTFAT)

# Références



N. Perraudin, V. Kalofolias, D. Shuman, P. Vandergheynst.  
*UNLocBoX : A MATLAB convex optimization toolbox for proximal-splitting methods.*  
arXiv:1402.0779, 2016.



P. L. Combettes, J.-C. Pesquet.  
*Proximal splitting methods in signal processing.*  
In : Fixed-Point Algorithms for Inverse Problems in Science and Engineering, 2011.



A. Beck, M. Teboulle.  
*A fast iterative shrinkage-thresholding algorithm for linear inverse problems.*  
SIAM Journal on Imaging Sciences, 2009.



N. Perraudin, J. Paratte, D. Shuman, V. Kalofolias, P. Vandergheynst, D. K. Hammond.  
*GSPBOX : A toolbox for signal processing on graphs.*  
ArXiv:1501.00147, 2014.



# Merci pour votre attention !

Des questions ?